

AMENDMENTS TO THE CLAIMS

Detailed Listing of All Claims 1-24:

1 (Currently amended). A method comprising:

receiving an initial code associated with a bytecode framework, the initial code including a reference to a referenced class of the bytecode framework;

converting the initial code to a converted code capable of execution on an intermediate language code framework;

executing the converted code on the intermediate language code framework;

detecting a need for the referenced class during execution of the converted code on the intermediate language code framework; and

~~loading the referenced class into memory accessible by the intermediate language code framework;~~

executing code for the referenced class on the intermediate language code framework.

2 (Original). The method of claim 1 wherein the initial code comprises an applet.

3 (Original). The method of claim 1 further comprising creating a reference type for the referenced class.

4 (Canceled).

1 5 (Currently amended). The method of claim 1 wherein the code for
2 the reference class comprises code associated with a bytecode framework and
3 further comprising converting the code for the referenced class ~~code~~ to a converted
4 reference class code capable of execution on the intermediate language code
5 framework.

6
7 6 (Currently amended). The method of claim 5 wherein the further
8 ~~comprising executing~~ executes the converted referenced class code on the
9 intermediate language code framework.

10
11 7 (Currently amended). A computer-readable medium storing computer-
12 executable instructions to detect a need for a referenced class, the referenced class
13 comprising code associated with a bytecode framework; to initiate loading of the
14 referenced class code into memory associated with an intermediate language code
15 framework; ~~and to initiate conversion of the~~ reference class code to a converted
16 reference class code capable of execution on the intermediate language code
17 framework; and to initiate execution of the converted reference class code on the
18 intermediate language code framework.

19
20 8 (Currently amended). A method comprising:
21 receiving an initial code associated with a bytecode framework, the initial
22 code including a reference to a referenced class, the referenced class comprising
23 referenced class code associated with the bytecode framework;
24 converting the initial code to a converted code capable of execution on an
25 intermediate language code framework;

1 executing the converted code on the intermediate language code
2 framework;
3 detecting a need for the referenced class during execution of the converted
4 code on the intermediate language code framework;
5 loading code for the referenced class code into memory accessible by the
6 intermediate language code framework; ~~and~~
7 converting the code for the referenced class code to a converted code for
8 the reference class code capable of execution on the intermediate language code
9 framework; and
10 executing the converted code for the reference class on the intermediate
11 language code framework.

12
13 9 (Original). The method of claim 8 wherein the initial code comprises an
14 applet.

15
16 10 (Canceled).

17
18 11 (Original). The method of claim 9 further comprising creating a
19 reference type for the referenced class.

20
21 12 (Currently amended). A user system comprising:
22 an applet class loader;
23 a converter; and
24 an intermediate language code framework, wherein the converter converts
25 code associated with a bytecode framework to a converted code capable of

1 execution on the intermediate language code framework and wherein the applet
2 class loader loads class files comprising class file code associated with the
3 bytecode framework, the converter converts the class file code to a converted class
4 file code capable of execution on the intermediate language code framework; and
5 the intermediate language code framework executes the converted class file code.

6
7 13 (Canceled).

8
9 14 (Canceled).

10
11 15 (Currently amended). The system of claim 12 wherein the
12 intermediate language code framework comprises a runtime engine.

13
14 16 (Currently amended). The system of claim 12 wherein the
15 intermediate language code framework comprises an application domain.

16
17 17 (Currently amended). The system of claim 12 wherein the
18 intermediate language code framework comprises a dynamic assembly.

19
20 18 (Canceled).

21
22 19 (Currently amended). A user system comprising:
23 reception means for receiving an initial code associated with a bytecode
24 framework, the initial code including a reference to a referenced class;
25

1 conversion means for converting the initial code to a converted code
2 capable of execution on an intermediate language code framework;

3 execution means for executing the converted code on the intermediate
4 language code framework;

5 detection means for detecting a need for the referenced class during
6 execution of the converted code on the intermediate language code framework;
7 and

8 conversion means for converting code for the referenced class to converted
9 code for the referenced class capable of execution on the intermediate language
10 code framework wherein the execution means comprises means for executing the
11 converted code for the referenced class~~load means loading the referenced class~~
12 ~~into memory accessible by the intermediate language code framework.~~

13
14 20 (Currently amended). The user system of claim 19 further comprising
15 ~~wherein the load means includes an applet class loader.~~

16
17 21 (Original). The user system of claim 19 wherein the execution means
18 includes a runtime engine.

19
20 22 (Original). The user system of claim 19 wherein the reception means
21 includes a browser.

22
23 23 (Original). The user system of claim 19 wherein the initial code includes
24 an applet.

1 24 (Currently amended). A method comprising:
2 receiving an applet associated with a bytecode framework, the applet
3 including applet code and a reference to a referenced class;
4 converting the applet code to an intermediate language code capable of
5 execution on an intermediate language code framework;
6 executing the intermediate language code on the intermediate language
7 code framework;
8 detecting a need for the referenced class during execution of the
9 intermediate language code on the intermediate language code framework; and
10 loading the referenced class into memory accessible by the intermediate
11 language code framework;
12 converting code for the referenced class into converted code for the
13 referenced class; and
14 executing the converted code for the referenced class on the intermediate
15 language code framework.
16
17
18
19
20
21
22
23
24
25